Functions

Functions

```
fn say_hi() {
   println!(“Hello there!”);
}
```

Functions

```
fn say_hi() {
   println!(“Hello there!”);
}
```

```
fn main() {
   say_hi();
}
```

Functions

```
fn say_hi() {
  println!(“Hello there!”);
}
```

```
fn main() {
  say_hi();
}
```

```
fn main() {
  for i in 1..6 {
    say_hi();
  }
}
```

Function parameters

Pass by value

```
fn main() {
   let mut name = "John";
   say_hello(name);
   println!("{}", name);
}

fn say_hello(name: &str) {
   println!("Hello {}", name);
}
```

Function parameters

Pass by reference

```
fn main() {
   let mut name = "John";
   say_hello(&mut name);
   println!("{}", name);
}

fn say_hello(name: &mut &str) {
   *name = “Alex”;
   println!("Hello {}", name);
}
```

Function parameters

Return values

```
fn main() {
   let mut name = "John";
   let greeting = say_hello(&mut name);
   println!("{}", greeting);
}

fn say_hello(name: &mut &str) -> String {
   let greeting = format!("Hello {}", name);
   return greeting;
}
```

Function parameters

The return statement can be omitted

```
fn main() {
   let mut name = "John";
   let greeting = say_hello(&mut name);
   println!("{}", greeting);
}

fn say_hello(name: &mut &str) -> String {
   let greeting = format!("Hello {}", name);
   greeting
}
```